

JERZY GAŹDZICKI

518.5 : 526.5 : 681.142

## Programy rozwiązywania zadań geodezyjnych na polskiej uniwersalnej maszynie cyfrowej UMC 1

### Wstęp

Dynamiczny rozwój współczesnej techniki obliczeniowej stwarza nowe, szerokie możliwości postępu w wielu dziedzinach nauki i techniki. Zastosowanie do prac obliczeniowych szybkich, uniwersalnych maszyn elektronowych pozwala na rozwiązywanie problemów dotychczas niedostępnych dla rachmistrzów wyposażonych w papier, ołówek i arytmometr. Umiejętnie wykorzystywane przez wysoko kwalifikowanych specjalistów maszyny elektronowe zastępują ludzi w trudnej pracy rachunkowej, przynoszą dodatnie efekty ekonomiczne, przyczyniają się do zwiększania produkcji oraz polepszenia jej jakości przez użycie bardziej skomplikowanych, lecz poprawniejszych naukowo metod obliczeń.

W Instytucie Geodezji i Kartografii przy współpracy z Politechniką Warszawską prowadzi się od kilku lat badania mające na celu zastosowanie nowoczesnych maszyn matematycznych do obliczeń geodezyjnych. Spośród poddanych badaniom maszyn na szczególną uwagę zasługuje uniwersalna maszyna elektronowa UMC 1. Maszyna ta została zbudowana w Zakładzie Konstrukcji Telekomunikacyjnych i Radiofonii Politechniki Warszawskiej. Opracowanie planów konstrukcyjnych maszyny oraz realizacja ich w stosunkowo krótkim okresie czasu jest poważnym, twórczym osiągnięciem zespołu pracowników naukowych Zakładu kierowanego przez prof. A. Kilińskiego. Obecnie (wrzesień 1961 r.) eksploatuje się już dwa egzemplarze tej maszyny, tzw. model laboratoryjny i egzemplarz prototypowy. Dalsze maszyny tego typu oraz w udoskonalonej wersji UMC 1 są w fazie budowy.

Wyniki przeprowadzonych badań, w szczególności wyniki osiągnięte w zakresie programowania i próbnych obliczeń, które stopniowo przyjmują charakter obliczeń produkcyjnych, wskazują na to, że maszyna UMC 1 jest odpowiednia dla celów geodezyjnych. W codziennej praktyce obliczeń geodezyjnych przeważają zadania dość proste pod względem rachun-

kowym, a przy tym nie zawsze występujące w sposób masowy. Do zadań tego rodzaju nie jest celowe ze względów ekonomicznych stosowanie maszyn elektronowych o wysokim stopniu sprawności, natomiast pozytywne wyniki można osiągnąć używając maszyny elektronowe należące do klasy maszyn małych i stosunkowo tanich — typu UMC 1. Przemyślane, oryginalne rozwiązania konstrukcyjne predestynują tę maszynę również do rozwiązywania dość dużych problemów rachunkowych. Przewiduje się, że w najbliższym czasie maszyna UMC 1 zostanie użyta do wyrównania metodą spostrzeżeń pośredniczących sieci triangulacji wypełniającej, obejmującej około 500 punktów wyznaczanych, co się wiąże z rozwiązaniem układu około 1 000 równań normalnych.

Artykuł niniejszy jest przeznaczony dla czytelnika posiadającego elementarne wiadomości z zakresu eksploatacji maszyn cyfrowych. W rozdziale 1 podaje się pobieżny opis maszyny UMC 1, w rozdziale 2 — pewne wiadomości z zakresu programowania na tej maszynie, zaś w rozdziale 3 — opis prac eksperymentalnych przeprowadzonych przez Instytut Geodezji i Kartografii. Wyniki tych prac pozwalają na zorientowanie się w możliwościach dalszych zastosowań maszyny UMC 1.

## 1. Opis maszyny UMC 1

### 1.1. System minus-dwójkowy

Konstrukcja uniwersalnej maszyny cyfrowej UMC 1 (rys. 1) oparta została na systemie o zasadzie  $-2$ . W systemie tym liczby przedstawiamy w postaci sumy całkowitych potęg zasady  $-2$ , przy czym korzystamy z dwóch tylko cyfr: 0,  $-1$ . Możemy napisać ogólnie dla liczby rzeczywistej  $\alpha$ :

$$\alpha = (a_N \dots a_1 a_0, a_{-1} \dots)_{-2} = \sum_{-\infty}^N a_i (-2)^i \quad (1)$$

gdzie  $a_i = 0$  lub  $-1$ , a  $N$  jest największym  $i$ , przy którym  $a_i = -1$ .

Weźmy dla przykładu następujący zapis liczby  $\alpha_1$  w systemie minus-dwójkowym (znak minus dla wygody pisano nad cyfrą 1):

$$\alpha_1 = (\bar{1} \bar{1} \bar{1} \bar{1})_{-2}$$

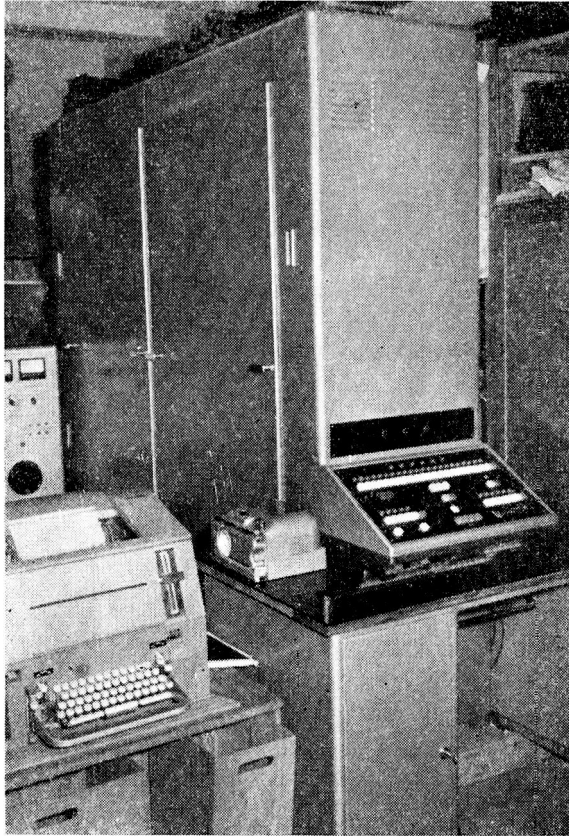
Przedstawimy tę liczbę w systemie dziesiętnym korzystając z wzoru (1):

$$\alpha_1 = -1 \cdot (-2)^3 - 1 \cdot (-2)^2 - 1 \cdot (-2)^1 - 1 \cdot (-2)^0 = 8 - 4 + 2 - 1 = 5$$

A oto drugi przykład:

$$\alpha_2 = (\bar{1} \bar{1} 0)_{-2} = -1 \cdot (-2)^2 - 1 \cdot (-2)^1 + 0 \cdot (-2)^0 = -4 + 2 = -2$$

Jak widać, w systemie minus-dwójkowym liczby ujemne są przedstawione w ten sam sposób jak liczby dodatnie, wobec czego nie zachodzi potrzeba zaopatrywania liczb znakiem plus lub minus, tak jak to ma miejsce w systemach o dodatniej zasadzie np. w systemie dziesiętnym czy też dwójkowym. Jednolite przedstawienie liczb dodatnich i ujemnych w systemie minus-dwójkowym jest istotną zaletą tego systemu.



Rys. 1. Ogólny widok maszyny UMC 1

Liczby dodatnie w systemie minus-dwójkowym mogą być łatwo odróżniane od liczb ujemnych. Można wykazać, że liczby ujemne posiadają  $N$  parzyste, zaś liczby dodatnie —  $N$  nieparzyste, np. dla liczby  $\alpha_1$ :

$$i = 3210$$

$$(111\bar{1})_{-2}$$

$N = 3$  jest nieparzyste, a zatem liczba jest dodatnia.

Wypiszemy kolejne potęgi zasady  $-2$ :

$$\begin{array}{cccccccc} i & \dots & 2 & 1 & 0 & -1 & -2 & -3 & \dots \\ (-2)^i & \dots & 4 & -2 & 1 & -1/2 & 1/4 & -1/8 & \dots \end{array}$$

Liczbę  $\alpha$  zgodnie z wzorem (1) możemy traktować jako sumę iloczynów potęg zasady przez odpowiadające tym potęgom cyfry: 0 lub  $-1$ . Niejednokrotnie wygodniej jest przy zapisie liczb w systemie minus-dwójkowym posługiwać się cyfrą 1 zamiast cyfrą  $-1$ , wówczas potęgi  $(-2)^i$  bierzemy ze zmienionym znakiem:

$$-(-2)^i \dots -4 \ 2 \ -1 \ 1/2 \ -1/4 \ 1/8 \dots,$$

a wzór (1) przyjmuje nieco inną postać:

$$\alpha = (a_N \dots a_1 a_0, a_{-1} \dots)_{-2} = - \sum_{-\infty}^N a_i (-2)^i \quad (2)$$

gdzie, tym razem,  $a_i = 0$  lub  $1$ .

Algorytmy działań arytmetycznych w systemie minus-dwójkowym są proste i w realizacji elektronicznej wymagają podobnego nakładu środków jak algorytmy systemu dwójkowego. Dla przykładu wykonamy sumowanie w systemie minus-dwójkowym dwóch liczb zapisanych w postaci (2).

$$(1111)_{-2} = 8 - 4 + 2 - 1 = 5$$

$$(1110)_{-2} = 8 - 4 + 2 = 6$$

$$(110101)_{-2} = 32 - 16 - 4 - 1 = 11$$

Przy wykonywaniu tego działania musimy pamiętać, że dwie jedyńki sumowane na  $i$ -tej pozycji cyfrowej dają nam przeniesienie równe  $-1$  na następną pozycję cyfrową, np.:

$$(1)_{-2} = -1$$

$$(1)_{-2} = -1$$

$$(\bar{1}0)_{-2} = -1 \cdot 2 = -2$$

Przeniesienie to równa się przeniesieniu dwóch jedynek na następane dwie (po  $i$ -tej) pozycje cyfrowe:

$$(\bar{1}0)_{-2} = (110)_{-2}$$

W maszynie UMC 1 liczby przedstawione są zasadniczo przy pomocy 36 pozycji cyfrowych:

$$a_1 a_0, a_{-1} a_{-2} \dots a_{-33} a_{-34},$$

jednakże w pamięci przechowuje się jedynie cyfry występujące po przecinku. Cyfry te pozwalają na przedstawienie liczb  $\alpha$  mieszczących się

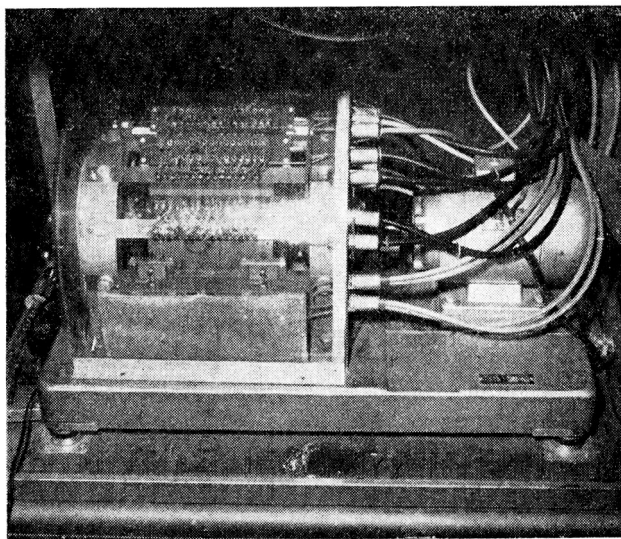
w następującym, niesymetrycznym względem zera przedziale:

$$-\frac{1}{3} < \alpha < \frac{2}{3}$$

Błąd wynikający z odrzucenia dalszych cyfr po przecinku nie przekracza  $0,4 \cdot 10^{-10}$ .

### 1.2. Pamięć i rejestry

Pamięcią maszyny UMC 1 jest bęben magnetyczny (rys. 2) wirujący z prędkością około 2700 obr./min. Bęben sporządzono z aluminium pokrytego cienką warstwą tlenku żelaza. W odległości około  $30\mu$  od powierzchni bębna znajdują się głowice odczytujące — zapisujące. Każdej głowicy odpowiada jedna ścieżka zapisu obejmująca 64 słowa. Bęben posiada 64 ścieżki, a całkowita jego pojemność wynosi  $64 \cdot 64 = 4096$  słów.



Rys. 2. Pamięć bębnowa

Maksymalny czas oczekiwania wynosi:  $1 \text{ min.} / 2700 = 0,022$  sek. Szybkość pracy maszyny określa się liczbą około 100 operacji na sekundę.

Na bębnie znajdują się również 3 technicznie wyodrębnione elementy pamięciowe — rejestr akumulatora **a**, rejestr mnożnika **m** i rejestr rozkazów **r**. Rejestry te mają długość jednego słowa.

Rejestr akumulatora spełnia zasadniczo rolę podobną do roli licznika rezultatów zwykłego arytmometru. Zawartość akumulatora może być przesuwana w lewo lub prawo, tzn. może być mnożona lub dzielona przez  $(-2)^n$ .

Rejestr mnożnika służy przede wszystkim do przechowywania mnożnika w operacji mnożenia, prócz tego rejestr ten spełnia istotną rolę w programowaniu.

W rejestrze rozkazów umieszczany jest rozkaz w czasie wykonywania tego rozkazu.

Oprócz wymienionych rejestrów bębnowych w maszynie UMC 1 jest jeszcze 5 innych rejestrów:

a) rejestr licznika rozkazów **l**, w którym przechowuje się bieżący adres wykonywanego rozkazu,

b) rejestr dalekopisu **d**, którego 5 pozycji odpowiada 5 dziurkom taśmy perforowanej, jest on łącznikiem między maszyną a dalekopisem,

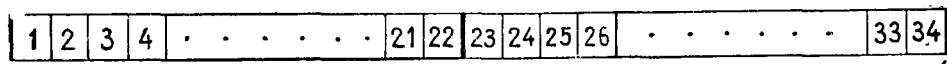
c) jednopozycyjowy rejestr warunku **w** — rejestr ten przechowuje informację uzyskaną z badania znaku lub nadmiaru liczby; rozkazy warunkowe wykonywane są zależnie od stanu rejestru **w**,

d) rejestr sterujący **s**, przechowujący część operacyjną rozkazu,

e) rejestr ścieżek **v**, przechowujący część adresu określającą numer ścieżki.

### 1.3. Operacje elementarne, rozkaz

Pamięć maszyny, rejestry i sumator połączone są tzw. bramkami, które pod wpływem sygnałów z urządzenia sterującego kierują pracą maszyny. Dla wykonania rozkazu należy otworzyć odpowiednie bramki na drodze przesyłania słowa z jednych elementów maszyny do innych, np. przesłanie słowa z pamięci do akumulatora wymaga otworzenia bramki odczytu z pamięci i bramki zapisu do akumulatora. Punktem centralnym przesyłania jest sumator. Sumator składa się z sumatora właściwego oraz dwóch, sterowanych również bramkami, układów zmieniających znak składników sumy; przy pomocy tych układów wykonuje się odejmowanie. Do sumatora podłączona jest także bramka „następnika” umożliwiająca wprowadzenie do rachunku jako trzeciego składnika jedynek na najmniej znaczącej pozycji, tj.  $1 \cdot 2^{-34}$ .



22 pozycje części operacyjnej rozkazu.

12 pozycji części adresowej rozkazu, 6 początkowych pozycji określa nr ścieżki, pozostałych 6 nr słowa na ścieżce.

Rys. 3

Rozkaz w maszynie UMC 1 składa się z części operacyjnej obejmującej 22 pozycje i części adresowej obejmującej 12 pozycji (rys. 3). Adresy mieszczą się w przedziale ( $-1365$ ,  $+2730$ ).

Każdej pozycji części operacyjnej odpowiada określona bramka w maszynie. Zależnie od stanu pozycji (0 lub 1) bramka otwiera się lub zamyka. Otworzenie bramki powoduje wykonanie odpowiedniej operacji elementarnej (tzw. mikrooperacji).

Działanie rozkazu w maszynie UMC 1 należy traktować jako wypadkową operacji elementarnych. Biorąc pod uwagę, że istnieją 22 operacje elementarne, a każda z nich może być wykonana lub też nie, będziemy mogli obliczyć ilość wszystkich możliwych rozkazów: jest ich  $2^{22}$ , tj. ponad 4 miliony. Pomimo faktu, że część spośród tych rozkazów nie ma sensu, istnieje możliwość utworzenia dla maszyny UMC 1 bardzo dużej ilości niezwykle cennych w programowaniu rozkazów — w odróżnieniu od większości eksploatowanych maszyn, posiadających stałą listę kilkadziesiątu rozkazów.

Umiejętne konstruowanie rozkazów maszyny UMC 1 i celowe ich używanie w programowaniu znacznie powiększa możliwości rachunkowe maszyny.

Poniżej wypisane zostały wszystkie pozycje części operacyjnej rozkazu:  
*Xo Do Adr Ro Lo Ns Od Ao Al Ap Nd Sg Wa Mn Xz Dz Az Lz Mo Mz Oa Rz*  
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22  
 Cyfra 1 na danej pozycji otwiera odpowiednią bramkę, cyfra 0 zamyka ją, wyjątek stanowią pozycje pierwsza i ostatnia, których bramki otwierają się przy cyfrze 0.

*Xo, Xz Do, Dz Ro, Rz Lo, Lz Ao, Az Mo, Mz*

oznaczają bramki odczytu i zapisu w odniesieniu do komórki pamięci o numerze X lub w odniesieniu do jednego z rejestrów:

**d, r, l, a, m.**

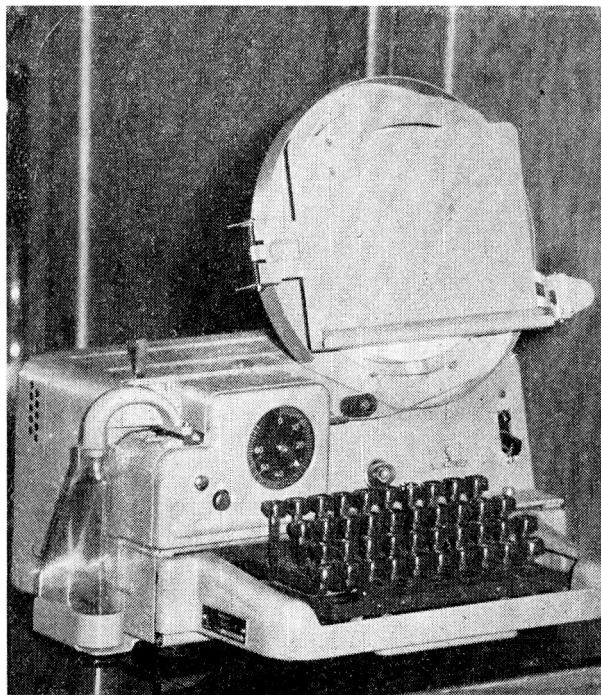
Pozostałe pozycje odpowiadają następującym operacjom elementarnym:

- Ns* — dodanie 1 na ostatniej pozycji cyfrowej w sumatorze,
- Od, Oa* — sterowanie układami zmiany znaku w sumatorze,
- Al, Ap* — przesuwanie zawartości akumulatora w lewo lub w prawo, tzn. mnożenie lub dzielenie przez  $-2$ ,
- Sg* — badanie znaku liczby wychodzącej z sumatora; w przypadku, gdy liczba jest większa od zera lub równa zeru do rejestru w zostaje przesłana cyfra 1, w przypadku, gdy liczba jest ujemna — cyfra 0,
- Nd* — badanie, czy liczba wychodząca z sumatora nie przekracza przedziału  $\left(-\frac{1}{3}, \frac{2}{3}\right)$ , jest to tzw. badanie nadmiaru; w przypadku stwierdzenia nadmiaru, do rejestru w zostaje przesłana cyfra 1, w przeciwnym przypadku — cyfra 0.

- Wa* — cyfra 1 na tej pozycji, dołączona do dowolnego rozkazu, uzależnia wykonanie tego rozkazu od wyniku badania znaku liczby lub nadmiaru; rozkaz wykonuje się tylko wtedy, gdy w rejestrze w znajduje się cyfra 0,
- Mn* — pozycji tej odpowiada cykl operacji dających w wyniku iloczyn,
- Adr* — cyfra 1 na tej pozycji dołączona do dowolnego rozkazu ogranicza działanie tego rozkazu tylko do części adresowej słowa.

#### 1.4. Urządzenia wejściowo-wyjściowe

Dane liczbowe i rozkazy wprowadza się do maszyny przy pomocy znormalizowanej taśmy dalekopisowej. Taśmę perforuje się przy pomocy dziurkarki (rys. 4) lub dalekopisu (rys. 6). Znaki cyfrowe i literowe przedstawione są na taśmie przy pomocy kombinacji otworów dziurkowanych w pięciu równoległych ścieżkach zapisu, przy czym tej samej kombinacji



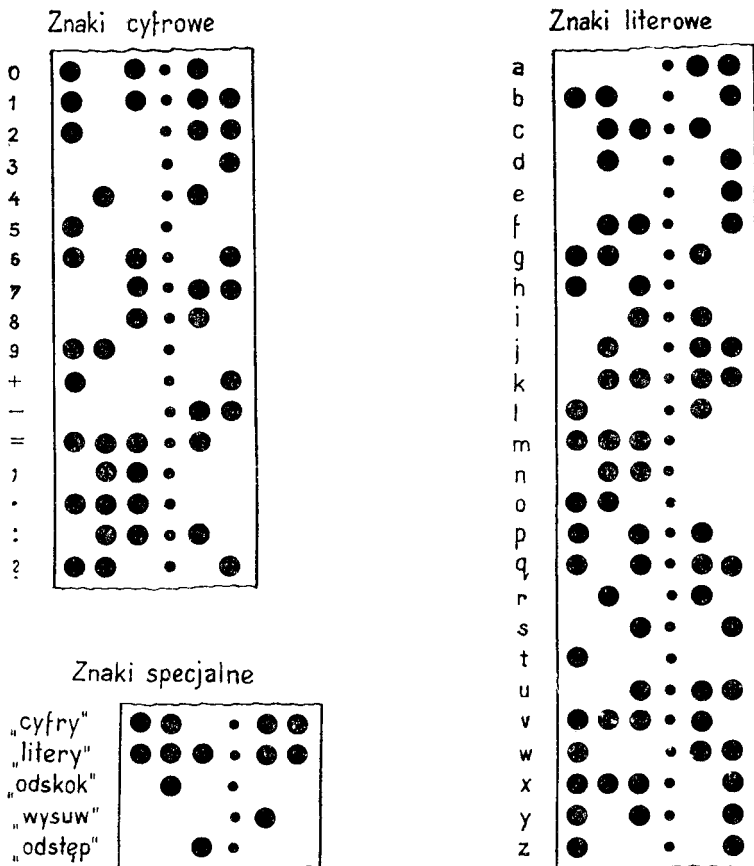
Rys. 4. Dziurkarka taśmy dalekopisowej

odpowiadają dwa różne znaki, np. literze „m” i znakowi „.” (kropka) odpowiadają otwory wydziurkowane w trzech pierwszych (licząc od lewej strony) ścieżkach zapisu. Dalekopis odczytujący taką kombinację drukuje „m”, jeśli uprzednio wydziurkowany został sygnał „literowy”, lub też



drukuje „”, jeśli uprzednio wydziurkowany został sygnał „cyfry”. Mniejsze otwory perforowane na taśmie nie są odczytywane i służą tylko do jej przesuwania w urządzeniach odczytujących.

Częściej używane znaki dalekopisowe



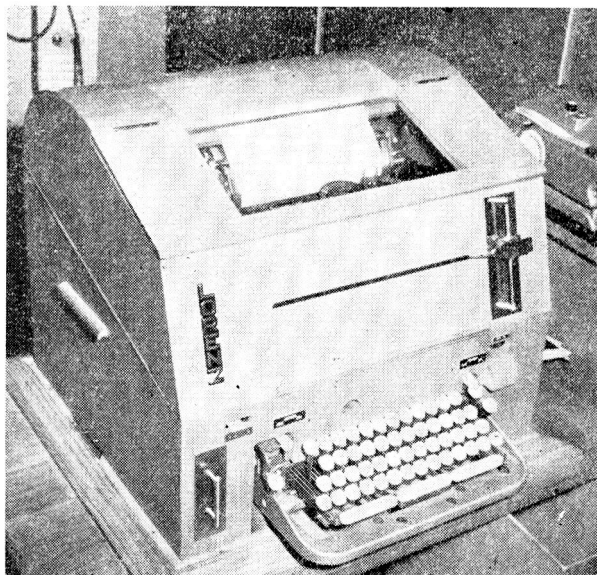
Rys. 5

Liczby i adresy dziurkuje się w systemie dziesiętnym, zaś części operacyjne rozkazów — przy pomocy znaków literowych, zgodnie z kodem opracowanym w Zakładzie Konstrukcji Telekomunikacyjnych i Radiofonii. Tłumaczenie zakodowanych rozkazów oraz przeliczenie danych liczbowych z układu dziesiętnego na minus-dwójkowy odbywa się w maszynie automatycznie przy pomocy specjalnego programu.

Po wydziurkowaniu taśmy wkładamy ją do urządzenia odczytującego, którym może być dalekopis lub specjalny czytnik (rys. 8). Urządzenie od-

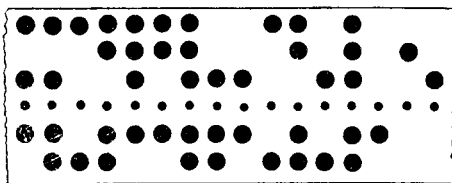
czytujące napotykać na dziurkę w taśmie przesyła do maszyny poprzez rejestr dalekopisu odpowiedni impuls prądu.

Wyniki rachunku drukowane są przez dalekopis w systemie dziesiętnym. Pośrednie wyniki rachunku mogą być dziurkowane na taśmie w postaci nadającej się do pownownego ich wprowadzenia do pamięci maszyny.



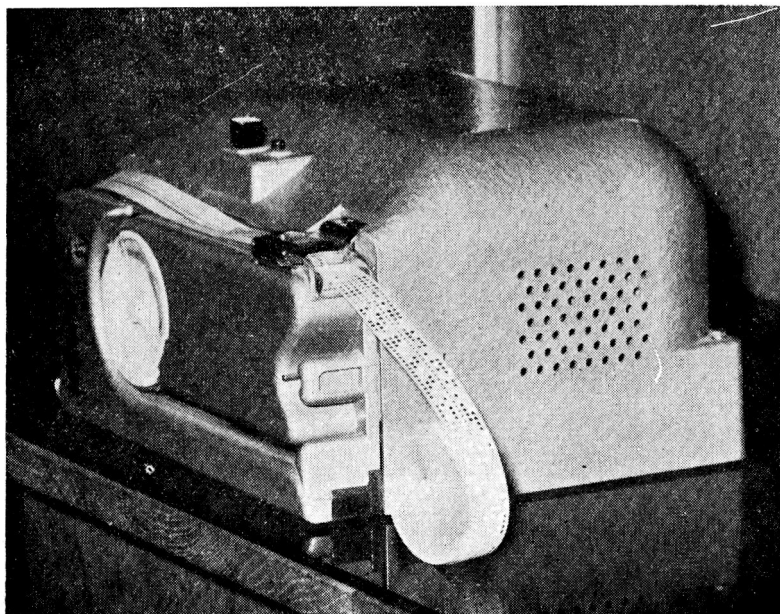
Rys. 6. Dalekopis

Szybkość pracy urządzeń wejściowo-wyjściowych wyniesi około 7 znaków na sekundę.



Rys. 7. Fragment taśmy dalekopisowej z wydziurkowanym na początku rozkazem  $s+87g$

Niewielkie ilości informacji mogą być wprowadzane do maszyny bezpośrednio, przy użyciu klawiatury dalekopisu.



Rys. 8. Czytnik taśmy dalekopisowej

## 2. Pewne wiadomości z zakresu programowania na maszynie UMC 1

### 2.1. Instrukcje

Konstruowanie rozkazów przy pomocy operacji elementarnych (mikrooperacji) wymaga szczegółowego zapoznania się z pracą maszyny. Dla ułatwienia programowania w Zakładzie Konstrukcji Telekomunikacyjnych i Radiofonii Politechniki Warszawskiej opracowano listę instrukcji składających się z jednego lub więcej rozkazów. Instrukcje te obejmują najczęściej spotykane w programowaniu operacje arytmetyczne, logiczne i inne, np. operacje obliczania wartości funkcji trygonometrycznych.

Poniżej podane zostały przykłady szeregu instrukcji, które dalej zostaną użyte do napisania programu.

Instrukcja	Treść instrukcji
$q.$	Umieść zero w akumulatorze.
$po+n.$	Prześlij słowo znajdujące się w akumulatorze do komórki pamięci o adresie $n$ (w skrócie: do miejsca $n$ ).
$r+n.$	Prześlij słowo znajdujące się w miejscu $n$ do akumulatora.

- $y + n$ . Do zawartości akumulatora dodaj słowo znajdujące się w miejscu  $n$ , wynik umieść w akumulatorze.
- $rc + n.pov + n$ . Prześlij słowo znajdujące się w miejscu  $n$  do akumulatora, dodaj do niego  $1 \cdot 2^{-34}$ , zwiększony w ten sposób o 1 adres prześlij z powrotem do miejsca  $n$ .
- $scx + n.fpov + 1$ , Zbadaj, czy ilość powtórzeń jest mniejsza od  $n$ ; instrukcję tę stosuje się w programach z tzw. pętlą.
- $cb + n$ . Jeżeli odpowiedź na pytanie zawarte w ostatniej instrukcji badania brzmi: a) „nie” — wykonaj następny w kolejności rozkaz, b) „tak” — wykonaj jako następny — rozkaz znajdujący się w miejscu  $n$ .
- $rt + n.xo + +.c + = 06$ . Podziel zawartość akumulatora przez liczbę znajdującą się w miejscu  $n$ , wynik umieść w akumulatorze.
- $wt.x.c + = 02$ . Wydrukuj liczbę znajdującą się w akumulatorze.
- $h$ . Stop, przerwij liczenie.
- $rt + n.xo + +.c + = 04$ . Pomnóż zawartość akumulatora przez liczbę znajdującą się w miejscu  $n$ , wynik umieść w akumulatorze.

Poszczególne rozkazy oddzielane są kropkami lub przecinkami.

Dwa rozkazy zasługują na specjalną uwagę. Są to tzw. rozkazy zewnętrzne:

- $s + ng$  Umieść następujące dalej słowa w kolejne miejsca pamięci, poczynając od miejsca  $n$ . Rozkaz ten piszemy zawsze na początku grupy liczb lub rozkazów wczytywanych do pamięci maszyny.
- $c + ng$  Start. Wykonuj rozkazy znajdujące się w kolejnych miejscach pamięci poczynając od rozkazu znajdującego się w miejscu  $n$ . Najpierw zostanie wykonany rozkaz z miejsca  $n$ .

(Uwaga. Powyższe instrukcje przedstawione są w sposób bardzo ogólny, a jednocześnie pobieżny. Podanie poprawniejszych opisów tych instrukcji wymagałoby szczegółowego omówienia działania poszczególnych rozkazów).

## 2.2. Liczenie w skali

Maszyna UMC 1 jest maszyną stało przecinkową; wszystkie 34 pozycje cyfrowe są w niej traktowane jako pozycje występujące za przecinkiem, a więc zarówno dane początkowe, jak i wyniki rachunku muszą się

mieścić w przedziale  $\left(-\frac{1}{3}, \frac{2}{3}\right)$ . Liczby tego przedziału, postaci  $\alpha \cdot 2^{-34}$ , zapisujemy w następujący sposób:

$$v + \alpha.$$

Powyższy kod stosuje się w szczególności do adresów, np.  $v-243$ . oznacza zapis adresu  $-243$ .

W pewnych przypadkach wygodniej jest liczyć z przecinkiem położonym nie przed pierwszą pozycją cyfrową, a np. między dziesiątą i jedenaścą pozycją cyfrową. Przy takim położeniu przecinka moglibyśmy operować liczbami w przedziale ograniczonym wartościami:

$$-\frac{1}{3} \cdot 2^{10} = -3,4 \cdot 10^2 \quad \text{oraz} \quad \frac{2}{3} \cdot 2^{10} = 6,8 \cdot 10^2.$$

Rachunek wykonywany w podanym przedziale nazywa się liczeniem w skali  $2^{-10}$ .

W Zakładzie Konstrukcji Telekomunikacyjnych i Radiofonii opracowano system liczenia w skalach  $2^{-2}$ ,  $2^{-4}$ ,  $2^{-6}$ ,  $2^{-8}$ ,  $2^{-10}$ ,  $\dots$ ,  $2^{-30}$ ,  $2^{-32}$ . Operator maszyny, który chce prowadzić obliczenie w wybranej przez siebie skali, przed wczytaniem danych liczbowych musi wprowadzić do maszyny 3 krótkie taśmy perforowane wchodzące w skład specjalnej biblioteki taśm perforowanych:

- a) taśmę określającą daną skalę,
- b) taśmę określającą ilość cyfr dziesiętnych za przecinkiem w liczbach wprowadzanych do maszyny,
- c) taśmę określającą ilość cyfr dziesiętnych za przecinkiem w liczbach drukowanych przez maszynę.

Maksymalna ilość cyfr za przecinkiem zależy od skali. Przy skali  $2^{-2}$  może być co najwyżej 9 cyfr za przecinkiem, przy skali  $2^{-20}$  — 4 cyfry, a przy skali  $2^{-32}$  — 0 cyfr.

Liczbę  $\alpha$ , która przez maszynę ma być interpretowana jako liczba w skali, zapisujemy w następujący sposób:

$$v + \alpha \text{ decm}$$

np. liczbę  $-10,15$  w skali  $2^{-12}$ , przy ustalonej ilości 4 cyfr za przecinkiem zapisujemy w postaci:

$$v - 101500 \text{ decm}$$

Instrukcje arytmetyczne podane w punkcie 2.1. odnoszą się do liczb zapisanych w skali.

## 2.3. Przykład

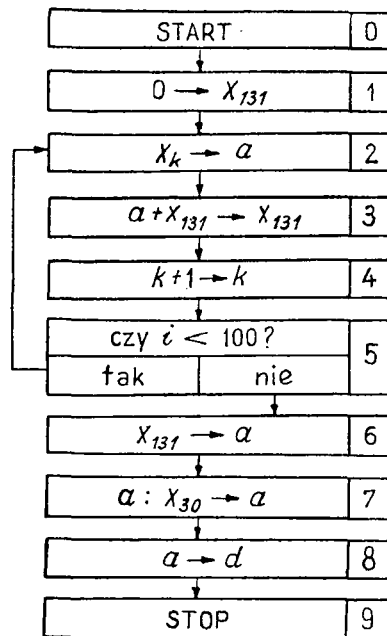
Weźmy pod uwagę proste zadanie obliczenia średniej arytmetycznej stu liczb:

$$X = \frac{\sum_{i=1}^{100} x_i}{100}$$

Ułożymy krótki program, przy pomocy którego zadanie to będzie mogło być rozwiązane.

Załóżmy, że liczby  $x_i$  umieszczone są już w miejscach pamięci: 31, 32, ... 130 oraz, że liczba 100 umieszczona jest w miejscu 30. Załóżmy dalej, że skala została dobrana tak, aby w trakcie liczenia nie powstał nadmiar.

Sporządzimy najpierw schemat blokowy (rys. 9), w skład którego wejdzie tzw. pętla obejmująca 2., 3., 4. i 5. klatkę schematu. Operacje występujące w tych klatkach będą wykonywane tyle razy ile jest liczb  $x_i$  do zsumowania, tj. 100 razy. \*)



Rys. 9

(Uwaga: Program opracowany został w sposób jak najbardziej uproszczony)

\*) Ogólne wiadomości z zakresu programowania można znaleźć między innymi w pracy [7].

Objaśnienia poszczególnych klatek schematu:

1. Umieść zero w miejscu pamięci 131, w miejscu tym będziemy umieszczali sumy liczb  $x_i$ :  $x_1 + 0$ ,  $x_1 + x_2$ ,  $x_1 + x_2 + x_3$ ,  $x_1 + x_2 + x_3 + x_4$  itd.
2. Prześlij liczbę z miejsca  $k$  do akumulatora; początkowo prześlemy liczbę  $x_1$  z miejsca 31, a więc będzie:  $x_{31} \rightarrow a$ .
3. Do zawartości akumulatora dodaj liczbę znajdującą się w miejscu 131, wynik umieść w miejscu 131.
4. Zwiększ o 1 adres w rozkazie 2, po wykonaniu tego rozkazu po raz pierwszy, otrzymamy rozkaz 2 w postaci:  $x_{32} \rightarrow a$ , potem  $x_{33} \rightarrow a$  itd.
5. Zbadaj, czy ilość powtórzeń w pętli jest mniejsza od 100, jeśli „tak” — przejdź do wykonania rozkazu 2, jeśli „nie” — wykonaj rozkaz 6.
6. Prześlij liczbę z miejsca 131 (będzie to już  $\sum_{i=1}^{100} x_i$ ) do akumulatora.
7. Podziel zawartość akumulatora przez liczbę znajdującą się w miejscu 30 (tj. przez 100), wynik umieść w akumulatorze.
8. Wydrukuj zawartość akumulatora.

Po tych dość szczegółowych objaśnieniach przystąpimy do przetłumaczenia symbolicznie przedstawionych w schemacie blokowym rozkazów na język podanych uprzednio instrukcji.

Program wprowadzimy do pamięci poczynając np. od miejsca 150.

	$s + 150g$
150	$q.$
151	$po + 131.$
152	$r + 31.$
153	$y + 131.$
154	$po + 131.$
155	$rc + 152.$
156	$pov + 152.$
157	$scx - 100.$
158	$fpov - 1,$
159	$cb + 152.$
160	$r + 131.$
161	$rt + 30.$
162	$xo + =.$
163	$c + = 06.$
164	$wt.$
165	$x.$
166	$c + = 02.$
167	$h.$
	$c + 150g$

Wprowadzenie rozkazu  $c + 150g$  jest sygnałem „startu”, maszyna rozpocznie liczenie wykonując jako pierwszy — rozkaz umieszczony w miejscu 150. Po skończeniu rachunku i wydrukowaniu średniej arytmetycznej zostaje wykonany rozkaz umieszczony w miejscu 167, jest to rozkaz „stop”.

### 3. Programy rozwiązywania zadań geodezyjnych

#### 3.1. Uwagi ogólne

W Instytucie Geodezji i Kartografii opracowano szereg programów rozwiązywania zadań geodezyjnych, między innymi programy: wyrównania sieci niwelacyjnych i grawimetrycznych, transformacji współrzędnych, rozwiązywania układów równań normalnych, obliczenia powierzchni ze współrzędnych, wcięcia w przód i wcięcia wstecz, wyrównania ciągu poligonowego itd. Przy doborze tematyki zadań z jednej strony uwzględniano bieżące potrzeby produkcji geodezyjnej, z drugiej strony kierowano się chęcią ukazania, w sposób możliwie pełny, możliwości rachunkowych maszyny.

Należy podkreślić, że dla maszyny uniwersalnej, jaką jest maszyna UMC 1, można opracować bardzo wiele programów dla tego samego problemu; programy te będą się różniły nie tylko zastosowaną metodą numeryczną, ale także wewnętrzną budową programu, organizacją występujących w nim pętli, sposobem wprowadzania danych i wyprowadzania wyników itd. Istotną trudność w programowaniu sprawia właśnie analiza różnorodnych wariantów programu, wybranie z nich wariantu optymalnego, najszybciej rozwiązującego określone zadanie rachunkowe. Dlatego też w biurach obliczeniowych wyróżnia się wysoko kwalifikowanych problemistów, którzy opracowują program od strony koncepcyjnej oraz rutynowanych programistów zajmujących się tłumaczeniem podanej koncepcji na język rozkazów. Problemista musi być w pełni zapoznany zarówno ze specyfiką metod numerycznych, stosowanych przy rozwiązywaniu danej klasy zagadnień, jak też z techniką programowania i właściwościami eksploatacyjnymi maszyny.

Tak więc każdy z opisywanych dalej programów jest jednym z wielu możliwych wariantów, wybranym z pewnych określonych względów. Niewątpliwie, niektóre z tych programów będą w przyszłości modyfikowane w celu osiągnięcia lepszych pod względem technicznym i ekonomicznym wyników.

Różnorodność opracowanych programów wskazuje na to, że maszyna UMC 1 może być użyta do rozwiązywania właściwie wszystkich zadań występujących w obliczeniach geodezyjnych. Jednakże, przy obecnych, dość wysokich kosztach amortyzacji i eksploatacji, celowe jest stosowanie tej maszyny tylko w pewnej klasie problemów obliczeniowych. Najogól-



niej rzecz biorąc, maszyna elektronowa stosowana w obliczeniach geodezyjnych daje wyniki korzystne pod względem ekonomicznym wówczas, gdy zadania są rozwiązywane w sposób masowy, a jednocześnie charakteryzują się dość dużą ilością operacji wykonywanych na niewielkiej ilości danych początkowych.

Przeprowadzone eksperymenty pozwalają na stwierdzenie szczególnie dobrych efektów ekonomicznych we wszelkiego rodzaju pracach transformacyjnych, w przeliczeniu współrzędnych Gaussa-Krügera z pasa danego na sąsiedni itp., jak również w pracach związanych z wyrównaniem sieci geodezyjnych: poligonowych, triangulacyjnych i niwelacyjnych. Należy przypuszczać, że pozytywne wyniki zostaną osiągnięte także i w innych działach obliczeń geodezyjnych. W najbliższym czasie podjęte zostaną prace przy programowaniu masowo występujących zadań fotogrametrycznych.

Dalszy rozwój automatyzacji obliczeń geodezyjnych napotyka na swej drodze na dwie podstawowe przeszkody. Jedną z nich jest fakt, że jak do tej pory żadna z instytucji geodezyjnych nie posiada jeszcze maszyny elektronicznej. Nie ulega kwestii, że szerokie zastosowanie współczesnej techniki obliczeniowej w geodezji może mieć miejsce tylko w oparciu o własną aparaturę. Istnieją poważne nadzieje, że w r. 1962 przeszkoda ta zostanie usunięta przez zainstalowanie maszyny UMC 1 w Instytucie Geodezji i Kartografii. Drugą przeszkodą jest niedostateczna ilość specjalistów w zakresie eksploatacji maszyn elektronicznych. Przewycięzenie związanych z tym trudności może nastąpić tylko i wyłącznie poprzez podjęcie systematycznego szkolenia.

W dalszym ciągu przedstawione zostaną opisy szeregu opracowanych już programów. Pierwsze dwa spośród tych opisów, odnoszące się do programów transformacji współrzędnych i wyrównania sieci niwelacyjnych, podane są przykładowo w sposób dość szczegółowy, następne zawierają tylko najważniejsze informacje o programach oraz wyniki obliczeń.

### 3.2. Program transformacji współrzędnych płaskich z jednoczesnym obliczeniem poprawek transformacyjnych metodą prof. Hausbrandta

#### 3.2.1. Opis programu

Program składa się z dwóch części. W pierwszej części realizuje się wzory transformacji współrzędnych płaskich:

$$\left. \begin{aligned} x_i &= x_0 + x'_i + x'_i z - y'_i t \\ y_i &= y_0 + y'_i + y'_i z + x'_i t \end{aligned} \right\} \quad (3)$$

$$i = 1, 2, \dots, N$$

w których oznaczono:

- $x_i, y_i$  — współrzędne obliczone w wyniku transformacji,  
 $x'_i, y'_i$  — współrzędne dane,  
 $x_0, y_0, z, t$  — współczynniki transformacji.

W drugiej części oblicza się poprawki do współrzędnych transformowanych  $x_i, y_i$  według wzorów:

$$v_{x_i} = \frac{\sum_{j=1}^n p_{i,j} V_{xj}}{\sum_{j=1}^n p_{i,j}} \quad (4)$$

$$v_{y_i} = \frac{\sum_{j=1}^n p_{i,j} V_{yj}}{\sum_{j=1}^n p_{i,j}}$$

gdzie:

$$p_{i,j} = \frac{\text{cnst}}{(x_i - X_j)^2 + (y_i - Y_j)^2}, \quad (5)$$

$X_j, Y_j$  — współrzędne punktów łącznych (nawiązania),

$V_{xj}, V_{yj}$  — poprawki współrzędnych punktów łącznych.

Wzory (4) opierają się na cennej metodzie prof. St. Hausbrandta, która znalazła obecnie szerokie zastosowanie w transformacji sieci triangulacji lokalnej na układ państwowy. Po zastosowaniu transformacji Helmerta powstaje problem usunięcia rozbieżności pomiędzy współrzędnymi państwowymi i transformowanymi punktów nawiązania (we wzorach (4) rozbieżności te oznaczono przez  $V_{xj}, V_{yj}$ ). Do niedawna używano w tym celu transformacji afinicznej, metoda prof. Hausbrandta [3] rozwiązuje ten problem w niewielkim stopniu zniekształcając wartości obserwacji. Jednocześnie okazało się, że wzory tej metody, traktujące w sposób jednolity wszystkie punkty transformowane w danej sieci, lepiej odpowiadają właściwościom eksploatacyjnym maszyn elektronowych, pomimo, że wymagają one wykonywania dużej ilości operacji arytmetycznych, a szczególnie — dużej ilości dzieleni przy obliczaniu wartości  $p_{i,j}$ .

W maszynie UMC 1 dzielenie jest operacją programowaną o stosunkowo długim czasie liczenia — 1,5 do 2 sek. Mając na względzie skrócenie czasu liczenia przeprowadzono analizę dokładnościową wzorów (4), która wykazała, że wartości  $p_{i,j}$  dla odległości

$$d_{i,j} = \sqrt{(x_i - X_j)^2 + (y_i - Y_j)^2}$$

większych od 10 km, z wystarczającą na ogół dokładnością można brać z tabelki:

$d_{i,j}$ w km	$p_{i,j} = \frac{1}{d_{i,j}^2}$
10,0—10,7	0,0094
10,7—11,5	0,0082
11,5—12,5	0,0070
12,5—13,8	0,0058
13,8—15,8	0,0046
15,8—18,9	0,0034
18,9—25,0	0,0022
25,0—50,0	0,0010

ułożonej przy założeniu, że  $d_{i,j} < 50$  km.

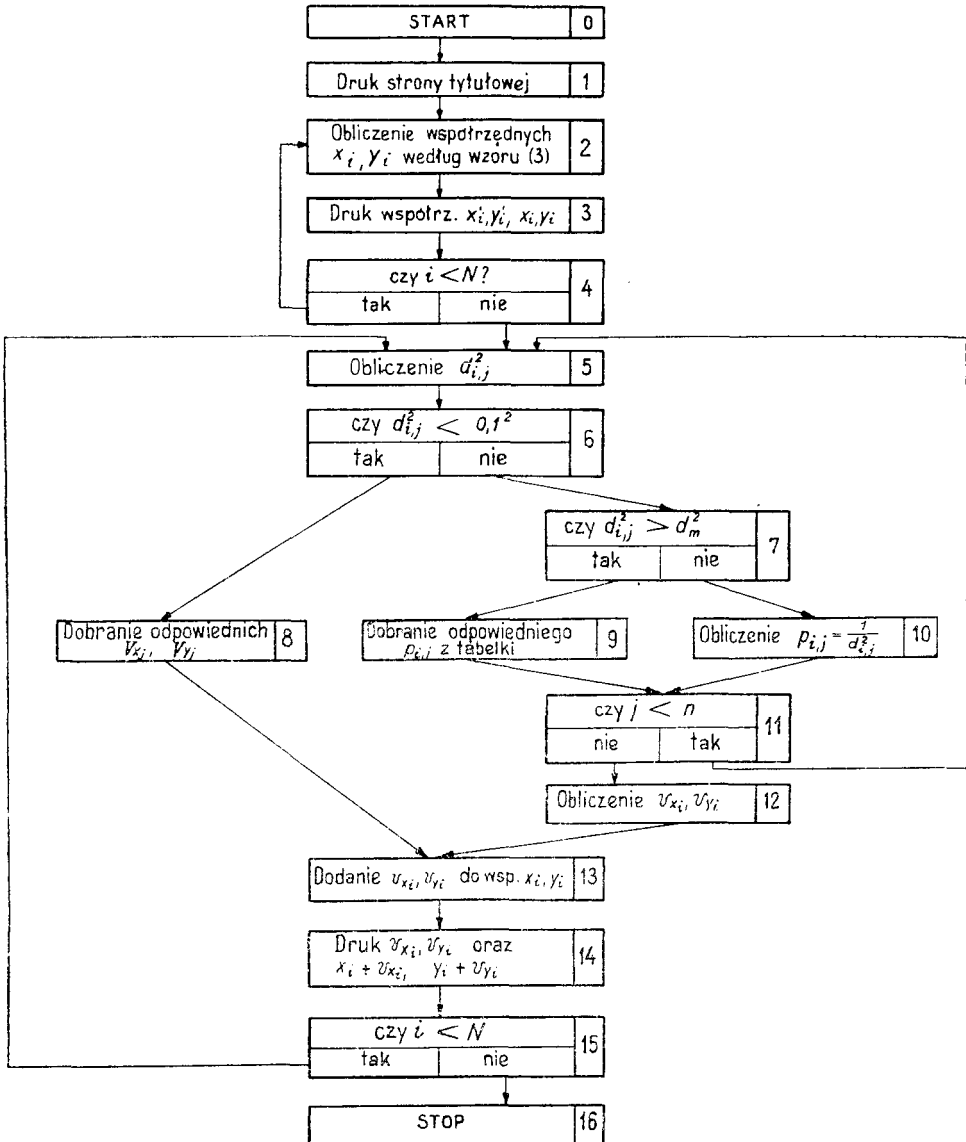
W związku z tym opracowano program, w którym wartości  $p_{i,j}$  dla  $d_{i,j} \leq d_m$  są obliczane według podanego wzoru, zaś dla  $d_{i,j} > d_m$  wartości te są przyjmowane zgodnie z powyższą tabelką. Wielkość  $d_m$ , zmienna w przedziale  $10 \div 50$  km, ustalana jest oddzielnie dla każdej sieci, zależnie od wielkości poprawek  $V_{x_j}, V_{y_j}$  oraz zależnie od ilości i położenia punktów nawiązania. Opisane postępowanie skróciło czas rachunku o kilkanaście do kilkadziesiąt procent.

W zbiorze współrzędnych przeliczanych według wzorów (3) i (4) znajdują się z reguły współrzędne punktów łącznych. Poprawki do tych współrzędnych są podane z góry ( $V_{x_j}, V_{y_j}$ ) i nie mogą być liczone przy pomocy wzoru (5) ze względu na to, że występuje wówczas w mianowniku odległość  $d_{i,j} = 0$ . Przypadek ten został uwzględniony w programie, w którym bada się, czy  $d_{i,j} < 0,1$  km, jeśli odpowiedź brzmi „tak” — przyjmuje się, że  $v_{x_i} = V_{x_j}$ ,  $v_{y_i} = V_{y_j}$ .

Program ten znalazł szerokie zastosowanie w pracach wykonywanych dla potrzeb Państwowego Przedsiębiorstwa Geodezyjnego.

### 3.2.2. Ogólny schemat blokowy (rys. 10)

Klatki 1, 2, 3, 4 obejmują rachunek współrzędnych transformowanych  $x_i, y_i$  oraz ich druk, łącznie z drukiem współrzędnych danych  $x'_i, y'_i$ . Ta część programu może być użyta niezależnie od pozostałej, w której dokonuje się obliczenia poprawek  $v_{x_i}, v_{y_i}$ , drukuje się te poprawki, a także oblicza się i drukuje współrzędne ostateczne  $x_i + v_{x_i}, y_i + v_{y_i}$ . Wyniki rachunku drukowane są w postaci 50 wierszowych tablic zajmujących powierzchnię jednej strony o formacie nieco mniejszym od A4.



Rys. 10

### 3.2.3. Dane techniczne programu

Maksymalna ilość jednocześnie transformowanych punktów: 500.

Maksymalna ilość punktów nawiazania: 50.

Dokładność rachunku: 1 mm.

Całkowita ilość miejsc pamięci zajęta przez program: 438.

Czas wprowadzania danych: około 5 sek na punkt.

Czas transformacji współrzędnych przy pomocy wzorów (3), łącznie z czasem drukowania współrzędnych: około 10 sek na punkt.

Czas liczenia poprawek (4) oraz współrzędnych ostatecznych łącznie z czasem drukowania poprawek i współrzędnych: przy  $10 < n < 20$  w przybliżeniu 25÷50 sek.

## 3.3. Program wyrównania sieci niwelacyjnych

### 3.3.1. Opis programu

Program służy do wyrównania sieci niwelacji geometrycznej lub trygonometrycznej metodą spostrzeżeń pośredniczących. Może on być również użyty do wyrównania sieci grawimetrycznych, a także do wyrównania dużych sieci poligonowych metodą punktów węzłowych.

Jako dane początkowe do maszyny wprowadza się:

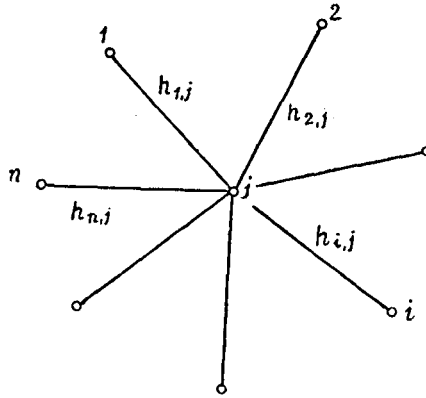
- liczbę obserwacji  $N$  oraz liczbę punktów wyznaczanych  $p$ ,
- wartości obserwacji,
- wartości wag obserwacji,
- wartości przybliżone rzędnych punktów wyznaczanych oraz wartości rzędnych punktów stałych,
- tzw. plan położenia punktów  $i$  obserwacji w sieci spełniający w maszynie rolę szkicu sieci; plan ten przyporządkowuje każdej obserwacji niwelacyjnej numery (adresy) punktów, pomiędzy którymi obserwacja ta została wykonana.

Po przeprowadzeniu obliczenia, maszyna drukuje:

- błąd średni pojedynczego spostrzeżenia —  $m_o$ ,
- sumę kwadratów poprawek obserwacyjnych mnożonych przez wagi tych obserwacji —  $[pvv]$ ,
- wyrównane wartości rzędnych punktów wyznaczanych —  $z_i$ ,
- błędy średnie tych rzędnych  $m_{z_i}$ ,
- wyrównane wartości obserwacji,
- poprawki do obserwacji.

Jak widać z powyższego, opisywany program wykonuje całość obliczeń związanych z wyrównaniem sieci niwelacyjnych.

Weźmy pod uwagę fragment sieci (rys. 11) przedstawiający punkt  $j$  oraz punkty  $1, 2, \dots, i, \dots, n$  połączone z punktem  $j$  obserwacjami  $h_{i,j}$  o wagach  $p_{i,j}$ .



Rys. 11

Wyrównane wartości rzędnych obliczane są w programie w oparciu o następujący wzór iteracyjny stosowany kolejno do wszystkich punktów wyznaczanych sieci:

$$z_j^{(r)} = \frac{\sum_{i=1}^n (z_i^{(r')} + h_{i,j}) p_{i,j}}{\sum_{i=1}^n p_{i,j}}, \quad (6)$$

gdzie:  $z_j^{(r)}$  —  $r$ -te przybliżenie rzędnej  $z_j$ ,

$z_i^{(r')}$  — przybliżenie rzędnej  $z_i$  obliczone po raz ostatni przed obliczeniem  $r$ -tego przybliżenia rzędnej  $z_j$ ; mogą zachodzić dwa przypadki:  $r' = r$  lub  $r' = r - 1$ .

Iterację uważamy za zakończoną, gdy dla wszystkich punktów mamy:  $z_j^{(R)} - z_j^{(R-1)} < \varepsilon$ , gdzie  $\varepsilon$  — wartość dostatecznie mała z punktu widzenia dokładności rachunku, ustalona z góry przed wyrównaniem sieci.

Poprawki obserwacji obliczamy według wzoru:

$$v_{i,j} = -h_{i,j} + z_j - z_i. \quad (7)$$

W przypadku gdy wyrównanie przeprowadza się metodą iteracji, pewne trudności nastęrcza podanie prawidłowej charakterystyki dokładnościowej rzędnych punktów wyznaczanych. Na ogół stosuje się wzór:

$$m_{z_j} = \pm \sqrt{\frac{\sum_{i=1}^n v_{i,j}^2 p_{i,j}}{(n-1) \sum_{i=1}^n p_{i,j}}}. \quad (8)$$

Obliczone przy pomocy tego wzoru błędy rzędnych znacznie różnią się od wartości błędów średnich. W związku z tym, opracowana została iteracyjna metoda obliczania błędów rzędnych, dająca wyniki bardzo bliskie wartościom błędów średnich. Metoda ta korzysta ze wzorów:

$$m_{z_j} = \pm m_0 \sqrt{Q_j}, \quad (9)$$

gdzie  $m_0$  błąd średni pojedynczej obserwacji,

$$m_0 = \pm \sqrt{\frac{[pvv]}{N-p}}, \quad (10)$$

zaś  $Q_j$  oblicza się metodą iteracji przy pomocy poniższego wzoru, który należy realizować kolejno w odniesieniu do wszystkich punktów sieci:

$$Q_j^{(r)} = \frac{1}{\sum_{i=1}^n \frac{p_{i,j}}{1 + p_{i,j} Q_i^{(r')}}}, \quad (11)$$

gdzie:  $Q_j^{(r)}$  —  $r$ -te przybliżenie współczynnika wagowego  $Q_j$ ,

$Q_i^{(r')}$  — przybliżenie współczynnika  $Q_i$ , obliczone po raz ostatni przed obliczeniem  $r$ -tego przybliżenia współczynnika wagowego  $Q_j$ ; mogą zachodzić dwa przypadki:  $r' = r$  lub  $r' = r - 1$ .

Jako pierwsze przybliżenia przyjmuje się wartości:

$$Q_j^{(0)} = \frac{1}{\sum_{i=1}^n p_{i,j}}. \quad (12)$$

Należy podkreślić, że procesy iteracyjne opisane wzorami (6) i (11) są szybko zbieżne. Uzasadnienie tych wzorów podane jest na końcu punktu 3.3.1.

Program opracowano w ten sposób, że po wykonaniu ustalonej z góry liczby  $r$  przybliżeń maszyna drukuje różnice  $z_j^{(r)} - z_j^{(r-1)}$  ( $j = 1, 2, \dots, N$ ) i zatrzymuje się, operator podejmuje wówczas decyzję, czy proces iteracyjny należy kontynuować, czy też nie. W pierwszym przypadku pisze rozkaz polecający maszynie wykonanie następnych  $r$  iteracji, wydrukowanie różnic między ostatnimi przybliżeniami i zatrzymanie się, w drugim — naciska guzik startu i maszyna prowadzi dalsze obliczenia wyrównawcze. W ten sposób operator ma możliwość kontrolowania przebiegu procesu iteracyjnego, co pozwala na wcześniejsze wykrycie błędów w danych początkowych. Ilość cykli po  $r$  przybliżeń jest dowolna.

Całkowitą ilość przybliżeń wykonywanych według wzoru (11) należy określić z góry, przy czym różnice między ostatnimi przybliżeniami współczynników wagowych nie są drukowane. Tego rodzaju uproszczenie

jest usprawiedliwione z jednej strony bardzo szybką zbieżnością procesu iteracyjnego, a z drugiej — mniejszymi wymaganiami odnośnie dokładności błędów średnich.

Program został zastosowany do celów produkcyjnych dla potrzeb Państwowego Przedsiębiorstwa Geodezyjnego i Przedsiębiorstwa Poszukiwań Geofizycznych. Przy ustalonej dokładności rachunku (1 mm dla sieci niwelacyjnych i 0,001 mgł dla sieci grawimetrycznych) wykonywano maksymalnie 12 iteracji drukując różnice:  $z_j^{(5)} - z_j^{(4)}$ ,  $z_j^{(9)} - z_j^{(8)}$ ,  $z_j^{(13)} - z_j^{(12)}$ .

\*

Poniżej podamy uzasadnienie wzorów (6) i (11).

Przypuśćmy, że znane nam są bezbłędne wartości rzędnych punktów 1, 2 ... n, otaczających punkt j. Przy tym założeniu, wartość rzędnej punktu j będziemy mogli traktować jako ogólną średnią arytmetyczną:

$$z_j = \frac{\sum_{i=1}^n (z_i + h_{i,j}) p_{i,j}}{\sum_{i=1}^n p_{i,j}}$$

W rzeczywistości postawione założenie nie jest słuszne, znamy tylko przybliżone wartości rzędnych  $z_i$ , wobec czego należy stosować postępowanie iteracyjne według wzoru (6). Można łatwo udowodnić, że postępowanie to odpowiada metodzie iteracji Gaussa-Seidla, użytej w odniesieniu do układu równań normalnych.

Przypuśćmy z kolei, że znane nam są wagi punktów 1, 2 ... n:  $p_1, p_2 \dots p_n$ . W tym przypadku rzędna punktu j przedstawimy jako ogólną średnią arytmetyczną wartości:  $z_i + h_{i,j}$  o wagach:

$$\left( \begin{array}{cc} 1 & 1 \\ p_i & p_{i,j} \end{array} \right)^{-1} = \frac{p_i p_{i,j}}{p_i + p_{i,j}}$$

Kładąc  $p_i \frac{1}{Q_i}$ , otrzymamy:

$$\frac{p_i p_{i,j}}{p_i + p_{i,j}} = \frac{p_{i,j}}{1 + p_{i,j} Q_i}$$

Waga rzędnej  $z_j$  równa się zatem:

$$p_j = \frac{\sum_{i=1}^n p_{i,j}}{\sum_{i=1}^n 1 + p_{i,j} Q_i},$$

a więc:

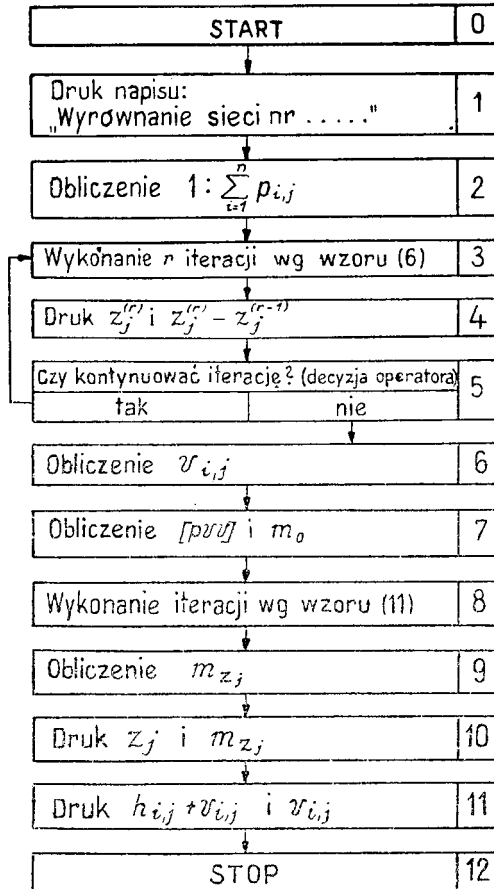
$$Q_j = \frac{1}{p_j} = \frac{1}{\sum_{i=1}^n \frac{p_{i,j}}{1 + p_{i,j} Q_i}}$$



Jednakże wagi rzędnych punktów 1, 2 ... n nie są nam znane, wobec czego należy stosować postępowanie iteracyjne według wzoru (11), przyjmując jako pierwsze przybliżenia wartości określone wzorem (12), wynikającym z wzoru (11) przy założeniu:  $Q_i = 0$ .

### 3.3.2. Ogólny schemat blokowy (rys. 12)

Schemat jest bardzo ogólny, nie uwidoczniono w nim wielokrotnie występujących pętli.



Rys. 12

### 3.3.3. Dane techniczne programu

Maksymalna ilość punktów w sieci: 100.

Maksymalna ilość obserwacji w sieci: 200.

Dokładność rachunku: może być dostosowana do potrzeb, zasadniczo obli-

czenia prowadzi się z dokładnością 1 mm.

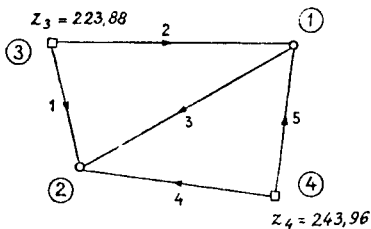
Całkowita ilość miejsc pamięci zajętych przez program: 429.

Czas wyrównania sieci składającej się z około 30 punktów wyznaczanych i około 100 obserwacji łącznie z czasem drukowania wyników:

40 ÷ 50 min. Czas wprowadzania danych dla takiej sieci wynosi około 10 min.

### 3.3.4. Przykład wyrównania małej sieci niwelacyjnej

Należy wyrównać sieć przedstawioną na rysunku.



Oznaczenia:

- punkt stały
- punkt wyznaczany
- 5 numer obserwacji z zaznaczeniem kierunku

Nr	Obserwacja $h_{i,j}$ w metrach	Waga $p_{i,j}$
1	18,600	1,7
2	1,200	1,3
3	17,600	1,5
4	-1,710	2,1
5	-19,020	1,1

Wyrównanie przeprowadzono opisaną metodą na maszynie UMC 1 oraz — dla kontroli — na arytmometrze, stosując metodę najmniejszych kwadratów w ujęciu wyznacznikowym.

Wyniki uzyskane na maszynie UMC 1 po 4 iteracjach:

$$z_1^{(5)} - z_1^{(4)} = 0,000$$

$$z_2^{(5)} - z_2^{(4)} = 0,000$$

$$[pvv] = 0,113 \quad m_0 = \pm 0,194$$

$$z_1 = 224,935 \quad m_{z_1} = \pm 0,103$$

$$z_2 = 242,404 \quad m_{z_2} = \pm 0,088$$

$$h_1 + v_1 = 18,524 \quad v_1 = -0,076$$

$$h_2 + v_2 = 1,055 \quad v_2 = -0,145$$

$$h_3 + v_3 = 17,470 \quad v_3 = -0,130$$

$$h_4 + v_4 = -1,556 \quad v_4 = +0,154$$

$$h_5 + v_5 = -19,025 \quad v_5 = -0,005$$

Wyniki, uzyskane na arytmetrze:

$$\begin{array}{ll}
 [pvv] = 0,113 & m_0 = \pm 0,194 \\
 z_1 = 224,935 & m_{z_1} = \pm 0,104 \\
 z_2 = 242,404 & m_{z_2} = \pm 0,089 \\
 h_1 + v_1 = 18,524 & v_1 = -0,076 \\
 h_2 + v_2 = 1,055 & v_2 = -0,145 \\
 h_3 + v_3 = 17,469 & v_3 = -0,131 \\
 h_4 + v_4 = -1,556 & v_4 = +0,154 \\
 h_5 + v_5 = -19,025 & v_5 = -0,005
 \end{array}$$

### 3.4. Program obliczenia powierzchni ze współrzędnych

Program oblicza dwukrotnie powierzchnię w oparciu o wzory Gaussa:

$$P = \frac{1}{2} \sum_{i=1}^N (x_i + x_{i+1})(y_{i+1} - y_i),$$

$$P = \frac{1}{2} \sum_{i=1}^N (y_i + y_{i+1})(x_i - x_{i+1}),$$

gdzie należy przyjąć  $y_{n+1} = y_1$ ,  $x_{n+1} = x_1$ .

Jako dane początkowe wprowadza się przy pomocy taśmy perforowanej:

- a) liczbę punktów załamania obwodnicy,
- b) współrzędne punktów załamania.

Czas dwukrotnego obliczenia i wydrukowania powierzchni przy około 30 punktach załamania wynosi około 2 minut. Dokładność rachunku 1 m<sup>2</sup>.

### 3.5. Program obliczenia współrzędnych przy pomocy wcięcia wstecz\*)

Program opracowano w oparciu o wzory prof. Hausbrandta. Dane początkowe, tj. współrzędne punktów stałych oraz kąty wprowadza się do maszyny przy pomocy klawiatury dalekopisu. Czas obliczenia i wydrukowania współrzędnych punktu wcinanego wynosi około 25 sek.

### 3.6. Program obliczenia współrzędnych przy pomocy wcięcia w przód

Program opracowano w oparciu o wzory prof. Hausbrandta. Dane początkowe, tj. współrzędne punktów stałych oraz kąty wprowadza się przy pomocy klawiatury dalekopisu. Czas obliczenia i wydrukowania współrzędnych punktu wcinanego wynosi około 20 sek.

\*) Programy wcięcia w przód i wstecz opracował mgr inż. Z. Zajac.

### 3.7. Program obliczenia współrzędnych punktów obustronnie nawiązanego ciągu poligonowego

Wielkościami danymi w tym programie są współrzędne punktów początkowego i końcowego oraz azymuty i długości boków. W wyniku obliczenia uzyskuje się: a) współrzędne wyrównane punktów ciągu, b) odchyłki  $f_x$  i  $f_y$ , c) dla kontroli — zniekształcenia na skutek wyrównania pomierzonych długości boków. Dane początkowe oraz wyniki pisane są przez maszynę w postaci tabulogramu o układzie odpowiadającym powszechnie stosowanemu formularzom rachunkowym. Dokładność rachunku: 1 mm. Czas rachunku: dla ciągu o 10 bokach około 5 minut.

### 3.8. Programy rozwiązywania symetrycznych układów równań liniowych

Prace przy programowaniu tego zadania nie są jeszcze zakończone, jednakże uzyskano już bardzo interesujące wyniki, np. rozwiązano układ 30 równań normalnych i obliczono odwrotność krakowianu współczynników tego układu w czasie niespełna 2 godzin. Opracowano także program rozwiązywania dużych układów równań metodą Pranis-Praniewicza. Program ten zostanie użyty do wyrównania wspomnianej we wstępie sieci triangulacji wypełniającej obejmującej 500 punktów. Wyniki prac w tym zakresie będą treścią oddzielnej publikacji.

## LITERATURA

- [1] Gaździcki J., Rzyśko M., Waśniewski J.: Zastosowanie mnożarki elektronicznej M22B do obliczeń masowych w geodezji, Przegląd Geodezyjny, Warszawa 1961.
- [2] Gaździcki J.: Zastosowanie maszyny UMC 1 do obliczeń geodezyjnych, Przegląd Geodezyjny, w druku.
- [3] Hausbrandt St.: W jaki sposób można by stosować transformację Helmerta przy rachunkowym opracowywaniu sieci triangulacyjnych nawiązanych do punktów stałych nie podlegających przesunięciu, Biuletyn Wojskowej Akademii Technicznej, Warszawa 1959.
- [4] Lazarkiewicz A., Balaśniński W.: A Simple Experimental Computer with Negative Basis, Mathematics of Computation, Washington, July, 1961.
- [5] Michalski T.: Triangulacja szczegółowa, Warszawa 1960.
- [6] Milewski J.: Elektroniczne maszyny cyfrowe, Przegląd Geodezyjny, Warszawa 1961.
- [7] Ostalski W., Gaździcki J.: Automatyczne maszyny cyfrowe i ich zastosowanie w geodezji, Prace Instytutu Geodezji i Kartografii, tom VII, Nr 2(16), Warszawa 1960.

Rękopis złożono w Redakcji we wrześniu 1961 r.

ЕЖЫ ГАЗЬДЗИЦКИ

## ПРОГРАММЫ РЕШЕНИЯ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ НА ПОЛЬСКОЙ УНИВЕРСАЛЬНОЙ ЦИФРОВОЙ МАШИНЕ УМС 1

### Резюме

В Кафедре Конструкции Телесвязи и Радиовещания Варшавского Политехнического Института построено универсальную электронную цифровую машину УМС 1. Машина УМС 1 принадлежит к классу малых машин. Она характеризуется скоростью работы порядка 100 операций в секунду и запоминающим устройством на магнитном барабане на 4096 чисел, состоящих из 34 двоичных разрядов. Конструкция машины основывается на системе счисления с основанием  $-2$ , что дает возможность однородного представления и положительных и отрицательных чисел. Для введения данных применено перфоленту, результаты печатаются при помощи рулонного телеграфного аппарата — телетайпа.

Институт Геодезии и Картографии при сотрудничестве Варшавского Политехнического Института начал исследования, имеющие целью проверить пригодность машины УМС 1 для геодезических вычислений. Разработано ряд программ, между ними программу решения систем линейных уравнений, программу уравнивания нивелирных и гравиметрических сетей, программу трансформационных вычислений, программу прямых и обратных засечек и т. д. Результаты экспериментальных вычислений при помощи этих программ представляются очень поощрительно. Особенно выгодным и полезным, с экономической точки зрения, является применение этой машины для решения массово вносимых задач, характеризующихся довольно большим числом операции производимых на небольшом числе исходных данных.

Программы уравнивания нивелирных сетей, трансформации и решения систем линейных уравнений были уже применены для массовых вычислений производственного характера. Проектируется установка машины УМС 1 в Институте Геодезии и Картографии и ее полное использование в разнородных областях геодезических вычислений.

*JERZY GAŹDZICKI*

## SOLUTION PROGRAMS OF GEODETIC PROBLEMS BY MEANS OF THE POLISH UNIVERSAL DIGITAL COMPUTER UMC 1

### S u m m a r y

A universal electronic digital computer has been built by the Chair of Radio — and Telecommunication Department of Polytechnical College in Warsaw. This computer belongs to a small computer class. It is characterized by a speed of an order of 100 operations per second and a capacity of drum-memory amounting to 4096—34 bit words. For positive and negative numbers being uniformly represented, the construction of the computer has been grounded on the position system with the base —2. To feed the informations a perforated tape has been applied and results are printed by means of a teleprinter.

The Institute of Geodesy and Cartography in co-operation with Polytechnic College in Warsaw started investigations to test the fitness of UMC 1 to geodetic computations. A number of programs has been worked out, and among them a program for solution of a linear equations system, a program for compensation of levelling and gravity nets, a program for transformation recalculations, programs for resection and intersection and so on. Experimental computations carried out by aid of those programs are very encouraging.

Application of the computer to solution of problems that occur in bulks and at the same time are characterized by a pretty large number of operations grounded on a small number of initial data is, from economical point of view, particularly advantageous.

Programs for compensation of levelling nets, transformations and solution of linear equation systems have just been used to mass computations of a productive character. It is planned to invest the computer UMC 1 in the Institute of Geodesy and Cartography, and to introduce it on a large scale to manifold fields of geodetic computations.